Multi-Layered Evaluation Using a Fusion of Metrics and LLMs as Judges in Open-Domain Question Answering

Rashin Rahnamoun and **Mehrnoush Shamsfard**
Shahid Beheshti University, Tehran, Iran
`rahnamounrashin@gmail.com` and `m-shams@sbu.ac.ir`

Abstract

Automatic evaluation of machine-generated texts, such as answers in open-domain question answering (Open-Domain QA), presents a complex challenge involving cost efficiency, hardware constraints, and high accuracy. Although various metrics exist for comparing machine-generated answers with reference (gold standard) answers, ranging from lexical metrics (e.g., exact match) to semantic ones (e.g., cosine similarity) and using large language models (LLMs) as judges, none of these approaches achieves perfect performance in terms of accuracy or cost. To address this issue, we propose two approaches to enhance evaluation. First, we summarize long answers and use the shortened versions in the evaluation process, demonstrating that this adjustment significantly improves both lexical matching and semantic-based metrics evaluation results. Second, we introduce a multi-layered evaluation methodology that combines different metrics tailored to various scenarios. This combination of simple metrics delivers performance comparable to LLMs as judges but at lower costs. Moreover, our fused approach, which integrates both lexical and semantic metrics with LLMs through our formula, outperforms previous evaluation solutions.

1 Introduction

The use of Large Language Models (LLMs) in various applications has increased significantly in recent years. These models are designed and optimized for a range of tasks and objectives, with evaluation being a key factor in understanding their performance. While human evaluation is considered the gold standard, it is both costly and time-consuming. As a result, many prefer automated evaluation methods, despite their higher error rates. These evaluations span different tasks and domains. In this paper, we focus on Open-domain Question Answering, where models are expected to generate appropriate answers to questions (Yang et al., 2019), a task whose evaluation poses unique challenges. Our goal is to develop an automated evaluation method that, with existing tools, can be applied across various scenarios with acceptable accuracy.

According to references (Zheng et al., 2023), and (Wang et al., 2023a), the approach of using LLMs, such as GPT-based models, as judges has shown remarkable performance compared to traditional methods. However, in real-world applications, many users may not want to rely on third-party services or expensive processes for evaluation. To address this, we propose a multi-layer evaluation methodology that incorporates both lexical-based metrics, such as exact match and ROUGE (Lin, 2004), and semantic-based metrics, including BERTScore (Zhang et al., 2020) and cosine similarity between vector embeddings, and others. In addition, large language models (LLMs) serve as judges to function as evaluation metrics, working in combination with these metrics to provide a comprehensive evaluation.

Previous works, such as (Kamalloo et al., 2023), (Adlakha et al., 2024), and (Wang et al., 2023a), relied solely on these metrics for evaluation, selecting the best one as the evaluator. However, our fused approach demonstrates that combining these metrics can improve accuracy. By applying our proposed formula, we show that this fusion of metrics in a multi-layer evaluation surpasses recent methods. Furthermore, extracting short answers from long model-generated responses and using them for evaluation significantly improves results for both lexical-based and semantic-based metrics.

In Layer 1, we apply highly accurate metrics that are effective at distinguishing between correct and incorrect data. The remaining data, after this filtering, is passed to Layer 2, where it is evaluated using metrics based on voting. For testing, we employed well-known datasets for Open-domain Question Answering evaluation, Natural Questions and TriviaQA which were recently used by (Chang

Proceedings of the 31st International Conference on Computational Linguistics, pages 6088–6104
January 19–24, 2025. ©2025 Association for Computational Linguistics

et al., 2024), (Li et al., 2024a), (Yang et al., 2024), (Li et al., 2024b) and (Cuconasu et al., 2024) works. We avoided custom metrics, instead relying on established metrics and both commercial and open-source LLMs to achieve results across different evaluation preferences.

Lexical-based metrics performed well after converting long generated responses into shorter forms. According to Kamalloo et al. (2023), the issue of low accuracy in lexical-based metrics was related to answer length; by addressing this, these metrics became reliable for filtering tasks. We also experimented with combining metrics based on varying requirements, budgets, and developmental needs. For low-budget solutions in Layer 2, we used lexical matching-based metrics, which performed similarly to GPT-3.5 Turbo as an evaluator.

Finally, by utilizing all available LLMs and metrics, and applying the optimal combination calculated by Eq. 7, we achieved a 3% improvement in the best automated evaluation results for Natural Questions, with 87% accuracy, and a 1% improvement in TriviaQA, with 97% accuracy in the short-answer form, for those seeking the most accurate results.

2 Related Work

2.1 LLM Evaluation

Researchers have explored various evaluation methods for large language models (LLMs) across different domains. For instance, An et al. (2024) tackled the issue of evaluating LLMs on tasks requiring long-context handling, while FineSurE Song et al. (2024) concentrated on text summarization performance. Another framework for assessing evaluation metrics was proposed by Xiao et al. (2023), and Balloccu et al. (2024) examined data leakage issues in closed-source LLMs.

Noteworthy evaluation techniques include zero-shot natural language evaluation through pairwise comparisons of LLM outputs (Liusie et al., 2024) and a method for assessing LLMs in conversational question answering tasks (Li et al., 2023). These studies underscore the complexity of automating LLM evaluation due to the diverse range of tasks and applications, highlighting the need for task-specific evaluation strategies.

2.2 Open-Domain Question Answering

In this paper, we focus on evaluating the Open-Domain Question Answering task, where the goal is for the model to generate accurate answers without additional context or clues about the correct answer. Evaluating this task is particularly difficult. As noted by Kamalloo et al. (2023), models often generate correct answers that may not match the "golden" reference answers exactly or may produce long, verbose responses that are hard to assess accurately. Their investigation into misjudgments in evaluation highlighted the absence of a fully reliable alternative to human evaluation.

Other work has explored evaluation for instruction-following models in question answering and highlighted the limitations of traditional metrics. (Adlakha et al., 2024) introduced a recall-based metric, while (Wang et al., 2023a) emphasized the importance of human evaluation and created a human-annotated dataset. Additionally, (Zheng et al., 2023) explored the use of LLMs as judges, suggesting it as a potential method for automating the evaluation process.

Furthermore, a new method introduced by Yona et al. (2024) proposes evaluating Open-Domain Question Answering models using a multi-granularity approach, providing a more nuanced assessment. Recently, many works have been proposed for achieving models or solutions for QA tasks, such as those by (Schimanski et al., 2024), (Chu et al., 2024), (Chen et al., 2024), (Huang et al., 2024), and (Faldu et al., 2024), which highlight that this task is challenging and underscore the importance of evaluating solutions.

2.3 Evaluation Metrics

Commonly used evaluation metrics can be categorized into three groups: lexical matching, semantic-based metrics, and the use of LLMs as judges. Lexical matching metrics include Recall and Precision, which compare tokens from reference and model-generated answers, as suggested by Adlakha et al. (2024). BLEU Score (Papineni et al., 2002) and ROUGE Score (Lin, 2004) evaluate text similarity using n-grams, while METEOR Score (Banerjee and Lavie, 2005) relies on the harmonic mean. Exact Match, on the other hand, requires a complete match with the reference answer. Another type of evaluation is based on semantic-based metrics, commonly used for QA tasks, as shown by Risch et al. (2021) with a bi-encoder-based metric that utilizes sentence transformers to calculate semantic similarity. Additionally, BERTScore, proposed by Zhang et al. (2020), measures token-level similarity. Another approach to evaluation leverages

LLMs as judges, where the models function as metrics for assessment, as explored by Zheng et al. (2023),Kamalloo et al. (2023),Adlakha et al. (2024) and Wang et al. (2023a).

3 Methodology

3.1 Problem Definition

As input, we consider a dataset D consisting of tuples $(q_i, r_i, m_i, \mathbb{H}_i)$, where $q_i \in Q$ denotes the i-th question, with Q representing the set of all possible questions. The corresponding reference (gold) answer for each question is given by $r_i \in R(q_i)$, where R is the set of all reference answers. The model-generated answer for the question q_i is $m_i \in M(q_i)$, with M being the space of all possible model-generated responses. Additionally, the human evaluation score for the pair (r_i, m_i) is represented by $\mathbb{H}_i \in H$, where H denotes the space of human evaluation scores.

Our objective is to develop an evaluation procedure $f_e : M \times R \to \mathbb{R}$, which computes an automated evaluation score $f_e(r_i, m_i)$ for the pair (r_i, m_i). The aim is to minimize the difference between the human evaluation score $\mathbb{H}_i$ and the automated score $f_e(r_i, m_i)$, which can be defined as finding:

$$\arg\min_{f_e} \sum_{i=1}^{n} |f_e(r_i, m_i) - \mathbb{H}_i|^2 . \quad (1)$$

Thus, our goal is to refine the evaluation procedure f_e such that it achieves the closest possible alignment with human evaluations $\mathbb{H}_i$.

3.2 Metric Functions

To calculate the evaluation score, we use a combination of different metrics $f_M(r_i, m_i)$, where each metric outputs a value within the range $[0, 1]$. Let $f_M(r_i, m_i)$ be a metric that outputs a value in $[0, 1]$, and let T_M be a threshold value. The evaluation of the model's response m_i is defined using the binary decision function $\phi(r_i, m_i)$ as follows:

$$\phi(r_i, m_i) = \begin{cases} 1, & \text{if } f_M(r_i, m_i) \geq T_M \\ 0, & \text{if } f_M(r_i, m_i) < T_M \end{cases} \quad (2)$$

In our experiments, we set the threshold $T_M = 0.5$ because human evaluation is represented in binary form, where 0 indicates incorrect and 1 indicates correct. To map the values to this binary format, we selected the midpoint as the threshold. These metrics go beyond just numerical values. Large Language Models (LLMs) can be used as evaluators, comparing the reference response with the model-generated response on a scale from 0 to 1.

3.3 Metric Scoring

The first step in selecting appropriate metrics for evaluation is to assess the accuracy of each metric relative to human judgment. To assign a score to each metric, we define the accuracy of the metric as the extent to which its evaluation aligns with human assessment. Let $f_M(r_i, m_i)$ represent the metric applied to the reference response r_i and the model's response m_i. The accuracy of the model is calculated by comparing the model's metric $f_M(r_i, m_i)$ against a threshold T_M. If the metric meets or exceeds the threshold, the result is treated as a boolean condition, which is then compared to the human evaluation boolean value $\mathbb{H}_i$. The accuracy is calculated as:

$$\text{Acc} = \frac{1}{|D|} \sum_{i \in D} I(\delta(f_M(r_i, m_i) \geq T_M) = \mathbb{H}_i) \quad (3)$$

The function $I()$ is an indicator function that outputs 1 if the condition inside it is true (i.e., if the comparison $f_M(r_i, m_i) \geq T_M$ aligns with the human evaluation $\mathbb{H}_i$), and 0 otherwise. Similarly, $\delta(f_M(r_i, m_i) \geq T_M)$ also converts the comparison into a boolean value, returning 1 if the condition $f_M(r_i, m_i) \geq T_M$ is satisfied, and 0 if it is not.

3.4 Evaluation Procedure

3.4.1 Extracting Short Answers

Before delving into the evaluation procedure, the first step is to calculate the accuracy of the metrics for both long model-generated answers and short extracted answers, as response length can impact the evaluation. The model's response may also be transformed from long to short form, depending on the specific question. To achieve this, we utilized a pre-trained RoBERTa-base model(Zhuang et al., 2021), fine-tuned on the SQuAD 2.0 dataset(Rajpurkar et al., 2018). This model, which is commonly used for extracting short answers from context , is one of the most popular models available through Hugging Face[1] for this task. We selected it due to its ease of use, and widespread availability to the public.

---

[1] `https://huggingface.co/deepset/roberta-base-squad2`

Dataset: Natural Questions

Question: Who plays the voice of johnny in sing?
Model Answer: Taron Egerton plays the voice of Johnny in Sing.
Extracted Short Answer: Taron Egerton
Reference Answer: Taron Egerton
Human Evaluation: True

Dataset: TriviaQA

Question: Who did Germany defeat to win the 1990 FIFA World Cup?
Model Answer: Germany defeated Argentina 1-0 in the 1990 FIFA World Cup Final.
Extracted Short Answer: Argentina
Reference Answer: Argentina
Human Evaluation: True

Figure 1: Two examples from our English datasets, illustrate the short answer extraction process output. It is important to note that human evaluation is based on the model's full answer.

Figure 1 shows examples of the input and output generated by this model. The accuracy of the long-response model and the accuracy of the extracted short-response model are calculated by comparing the model's metric $f_M(r_i, m_i)$ against a threshold T_M.

Although human evaluation focuses on long-form answers, converting the model's output into short-form answers may introduce errors for metrics that depend on short-form responses. However, in certain situations, this conversion can enhance the performance of specific metrics, despite the potential for occasional inaccuracies.

3.4.2 Threshold-Based Filtering (Layer 1)

The first layer of the evaluation procedure involves selecting highly accurate metrics that can effectively filter relevant data within their respective domains. In the filtering procedure, the goal is to select appropriate metrics that achieve high accuracy across various evaluation cases, aiming for metrics that can achieve a high accuracy rate, surpassing a threshold, such as 97%. To achieve this, we compute the accuracy using our formula from Eq. 3 for both long-form and short-form model-generated answers.

After selecting and sorting highly accurate metrics $\{M_1, M_2, \ldots, M_n\}$ from Eq. 3, the evaluation procedure begins. The first metric, M_1, is applied to filter the relevant data. The subset of data filtered by M_1 is represented as:

$$D_1 = \{d_i \in D \mid \phi_{M_1}(r_i, m_i) = 1\} \tag{4}$$

where $\phi_{M_1}(r_i, m_i)$ represents the metric value.

If $f_M \geq T_M$, the corresponding data will be filtered. The remaining data that is not filtered by M_1 is then passed to the next metric, M_2, and this process is repeated for each subsequent metric:

$$D_{k+1} = \{i \in (D \setminus D_k) \mid \phi_{M_{k+1}}(r_i, m_i) = 1\} \tag{5}$$

where D_k is the subset of data filtered by the previous metric M_k.

This iterative procedure continues until the final metric M_n is applied. The remaining data after filtering by all metrics in the layer are represented by:

$$D_{\text{remaining}} = D \setminus \bigcup_{k=1}^{n} D_k \tag{6}$$

where $D_{\text{remaining}}$ denotes the data that were not filtered by any of the metrics in Layer 1. These remaining data will be forwarded to Layer 2 for further evaluation.

3.4.3 Voting-Based Evaluation (Layer 2)

The remaining data from Layer 1 is evaluated using a voting mechanism. In this layer, most of the existing metrics, along with the remaining data, are assessed. For the evaluation, we employ the following our formula Eq. 7, which demonstrates the method for selecting appropriate metrics.

The first term in the formula, $Acc(f_{\mathbf{M}_i})$, represents the accuracy of each metric based on available human evaluations. The second term reflects the correlations between the metrics. To ensure that the metrics selected by the voting mechanism do not exhibit high correlations, we introduce a correlation threshold Eq. 8, denoted by Θ_{min} and Θ_{max}. In our experiments, the lower bound Θ_{min} is set to 0.6, while the upper bound Θ_{max} is set to 0.9. This constraint ensures that highly accurate metrics do not have low and very high correlations with one another. The reasons why we have chosen these numbers, along with their details, are provided in the Appendix A.

We computed correlations using Spearman (ρ_s), Pearson (ρ_p), and Kendall Tau (τ) (Kendall, 1945) , setting $k = 3$ in this formula. The third term of the formula accounts for the comparison between metric correlations and human judgment, following the method used for comparing the metrics with human judgment in (Liu et al., 2023). For simplicity, we have set β, λ and γ to 1. However, these coefficients could be adjusted to reflect the relative importance of each term. Further details on the use and application of this formula are provided in the Appendix B. In voting-based evaluation, the number of metrics must be odd because values above or below T_M represent a "yes" or "no" vote, respectively. For a final evaluation, it is necessary to avoid ambiguous results, which can occur when using an even number of metrics, as it may lead to indecisive outcomes. The formula for selecting the metrics in the voting-based evaluation is given by:

$$
\begin{aligned}
S = \arg \max_{S \subset \mathbf{M}, |S| \text{ odd}} & \Bigg[\beta \Big[\sum_{f_{\mathbf{M}_i} \in S} Acc(f_{\mathbf{M}_i}) \Big] \\
& - \lambda \left(\sum_{\substack{f_{\mathbf{M}_i}, f_{\mathbf{M}_j} \in S \\ i<j}} \frac{1}{k} \sum_{l=1}^{k} \rho_l(f_{\mathbf{M}_i}, f_{\mathbf{M}_j}) \right) \\
& + \gamma \left(\sum_{f_{\mathbf{M}_i} \in S} \frac{1}{k} \sum_{l=1}^{k} \rho_l(f_{\mathbf{M}_i}, \mathbb{H}_i) \right) \Bigg]
\end{aligned} \tag{7}
$$

Here, S denotes the set of selected metrics, where the number of selected metrics must be odd, $Acc(f_{\mathbf{M}_i})$ represents the accuracy of the metric $f_{\mathbf{M}_i}$, and $\rho_l(f_{\mathbf{M}_i}, f_{\mathbf{M}_j})$ is the correlation between metrics $f_{\mathbf{M}_i}$ and $f_{\mathbf{M}_j}$. The term $\rho_l(f_{\mathbf{M}_i}, \mathbb{H}_i)$ captures the correlation between metric $f_{\mathbf{M}_i}$ and human judgment $\mathbb{H}_i$.

The constraint on the correlations between metrics is expressed as:

$$
\begin{gathered}
\Theta_{\min} \leq \frac{1}{k} \sum_{l=1}^{k} \rho_l(f_{\mathbf{M}_i}, f_{\mathbf{M}_j}) \leq \Theta_{\max} \\
\text{for all } f_{\mathbf{M}_i}, f_{\mathbf{M}_j} \in S
\end{gathered} \tag{8}
$$

This constraint ensures that the selected metrics exhibit correlations within the predefined range $[\Theta_{\min}, \Theta_{\max}]$, thereby avoiding highly correlated metrics in the final selection.

The voting system aggregates the results from various metrics to produce a final decision. the voting mechanism is represented by the following equation:

$$
V_{\text{evaluation}} = \sum_{f_{\mathbf{M}_i} \in S} v(f_{\mathbf{M}_i}) \tag{9}
$$

Here, S denotes the set of selected metrics from Eq. 7, and $v(f_{\mathbf{M}_i})$ corresponds to the vote provided by the metric $f_{\mathbf{M}_i}$. The final result of evaluation, $V_{\text{evaluation}}$, is the simple sum of the votes from the selected metrics. The evaluation is based on these aggregated results.

4 Experiments

Following Wang et al. (2023a) and Yona et al. (2024), we used the TriviaQA and Natural Questions datasets, both popular benchmarks in the open-domain QA task ,to evaluate our automated evaluation methodology. Specifically, our objective is not to evaluate and compare different models on the same tasks, but to develop an efficient automated evaluation method with an acceptable error rate. To address this, we tested our automatic evaluation methodology on the model-generated answers discussed in Section 3, in order to find solutions to our problem, as formally defined in Eq. 1.

4.1 Datasets

Following Adlakha et al. (2024), Wang et al. (2023a) and Kamalloo et al. (2023), we used the TriviaQA (Joshi et al., 2017) and Natural Questions (Kwiatkowski et al., 2019) datasets, both popular benchmarks in the Open-Domain QA task and commonly used in Wang et al. (2023b), Fang et al. (2022),Izacard and Grave (2021) and Petroni et al. (2021), to evaluate our automated evaluation methodology. We utilized filtered versions of these datasets from (Wang et al., 2023a), excluding question-answer pairs with answers that were no longer suitable, such as those whose answers had changed over time.

Natural Questions. Natural Questions includes real user queries submitted to Google Search and answers sourced from Wikipedia, as annotated by human evaluators. From the filtered and model-generated responses of this dataset, we randomly selected 250 unique question-answer pairs from (Wang et al., 2023a), which were evaluated by five models: FiD, GPT-3.5, ChatGPT-3.5, ChatGPT-4, and NewBing. Human reviewers classified the responses as true, false, or improper, resulting in 1,088 valid pairs from an initial 1,250.

TriviaQA. TriviaQA, a reading comprehension dataset, we randomly selected 250 unique question-answer pairs from (Wang et al., 2023a). These were also evaluated by the same five models and reviewed by humans, leading to 1,245 valid pairs from an initial 1,250 after removing improper responses.

Both datasets, which include human annotations, were used from (Wang et al., 2023a), and the preparation steps are also explained in it.

4.2 Evaluation Methods

We applied widely used evaluation methods (Wang et al., 2023a) and our own custom approach with various configurations to achieve high accuracy compared to human judgments.

Lexical Matching. Lexical matching metrics are commonly used for model evaluation. These metrics compare reference answers with generated text but often perform poorly when there is no exact reference answer in generated answers or with long responses according to Kamalloo et al. (2023). This includes Exact Match, which requires an exact match with the reference answer; BLEU Score (Papineni et al., 2002) and ROUGE Score (Lin, 2004), which use n-grams to compare text according to their formulas; and METEOR Score (Banerjee and Lavie, 2005), which is based on the harmonic mean. Additionally, Word Matching is a custom metric that identifies matching words between the reference and generated text and calculates the accuracy percentage. Recall and Precision metrics, based on tokens from the reference and model-generated answers, were also used, as proposed by Adlakha et al. (2024).

Semantic Based. These scores focus on the semantics of text rather than finding matches. We employed BERTScore(Zhang et al., 2020), which uses token similarity through contextual embeddings. Additionally, we used BERT-based uncased embeddings combined with cosine similarity for evaluation. We also utilized the all-MiniLM-L6-v2 [2] model, a popular Hugging Face sentence transformer that operates using cosine similarity.

LLMs as Judges. Recently, strong LLMs used as judges have shown impressive results correlated with human evaluations. We compared the performance of different LLMs, including Llama 3.1 8B and Llama 3.1 70B (Dubey et al., 2024), both of which demonstrated excellent performance among open-source models. Additionally, GPT-4-o and GPT-3.5 Turbo(OpenAI et al., 2024) also performed very well in these evaluations. The prompts for LLMs to act as metrics are provided in Appendix C.

Our Method Setups. We explored different metrics tailored to specific needs. The first setup uses simple, widely-used lexical matching metrics that do not require third-party connections or powerful hardware, offering a cost-effective solution. The second setup combines these simple lexical matching metrics with semantic-based ones for semantic similarity checking, which are publicly available and require minimal hardware.

The third setup builds on previous lexical matching and semantic-based metrics by incorporating Llama 3.1 (70B and 8B), an open-source model known for strong performance. The fourth setup uses only Llama 3.1 8B to accommodate the hardware limitations of running the 70B model locally. The fifth setup relies solely on large language models (LLMs), appealing to users preferring third-party APIs without the need for development complexity. The sixth setup simplifies the process by using all metrics with just one LLM using Eq. 7. In the seventh setup, we applied all the metrics described to find the most suitable ones for our configuration, which required some human-annotated data for optimal performance using our formula from Eq. 7. Lastly, the eighth setup is the default, using the most commonly used metrics selected via Eq. 7 without customizing them, as we did not have access to human-annotated data.

4.3 Results

We applied commonly used automated evaluation methods, as outlined in Section 4.2, to assess the accuracy of model-generated answers against human judgments on the Natural Questions and TriviaQA datasets in Section 4.1. In some cases, we provided gold-standard answers to the models (denoted by "(Gold)") and compared the results. Short answers were extracted from model responses, as described in Section 3.3, and evaluated using different metrics for both long and short-form answers.

The original responses were in long form, but to further investigate the results, short answers were extracted and evaluated, which are detailed in Table 1.

4.4 Discussion and Analysis

Table 1 shows that, although Adlakha et al. (2024) demonstrated that commonly used lexical matching metrics perform poorly in open-domain QA, our results suggest otherwise. After applying our methodology, which is explained in Section 3.3, where we converted the model-generated responses into shorter forms, we observe significant improvements in the accuracy of lexical matching metrics. This change leads to more than a 60% improvement in the accuracy of lexical matching metrics. Additionally, the results indicate over a 40% improve-

---

[2] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Metric	Natural Questions		TriviaQA	
	Acc_{long}	$\text{Acc}_{\text{short}}$	Acc_{long}	$\text{Acc}_{\text{short}}$
Exact Match	0.38	0.67	0.21	0.82
BLEU Score	0.36	0.57	0.20	0.64
METEOR Score	0.42	0.79	0.30	**0.93**
ROUGE-2	0.38	0.57	0.25	0.40
ROUGE-L	0.42	0.72	0.29	0.56
Word Matching	0.71	0.81	0.50	0.91
Precision	0.41	0.79	0.34	**0.93**
Recall	0.81	**0.82**	0.59	**0.93**
BERT Score	0.50	0.77	0.38	0.88
Sentence Transformer	0.57	**0.84**	0.55	**0.94**
BERT Embedding	0.75	0.81	0.49	0.92
GPT-4-o (Gold)	0.84	**0.83**	**0.96**	**0.96**
GPT-4-o	0.76	0.74	0.89	0.89
GPT-3.5 Turbo (Gold)	0.82	0.80	0.91	0.90
GPT-3.5 Turbo	0.73	0.73	0.83	0.81
Meta-Llama 3.1 70B (Gold)	**0.85**	0.82	**0.96**	0.95
Meta-Llama 3.1 70B	0.72	0.71	0.84	0.83
Meta-Llama 3.1 8B (Gold)	0.77	0.79	0.72	0.65
Meta-Llama 3.1 8B	0.68	0.67	0.80	0.77

Table 1: The table compares the accuracy of various evaluation metrics for long and short answers from the Natural Questions and TriviaQA datasets. These metrics include lexical-based, semantic-based methods, and LLMs as judges. Results are split based on whether the LLMs had access to gold (reference) answers or not. The best results in each group are bolded, while the overall highest accuracy for each dataset is both bolded and underlined.

Evaluation Setup	Natural Questions	TriviaQA
1.Lexical Matching	0.81	0.92
2.Lexical Matching + Semantic-Based	0.83	0.93
3.Lexical Matching + Semantic-Based + Llama 3.1 All	0.86	0.96
4.Lexical Matching + Semantic-Based + Only Llama 3.1 8B	0.85	0.94
5.Only LLMs	0.82	0.96
6.Metrics Scoring Calculation + Only One LLM	0.85	0.96
7.Metrics Scoring Calculation	0.85	**0.97**
8.Metrics Scoring Calculation Default	**0.87**	**0.97**

Table 2: This table presents the accuracy of our different methodological setups, as explained in Section 4.2, for Natural Questions and TriviaQA datasets separately. Metrics selection calculation is described in Eq. 7. The results are based on short answer extraction.

ment in the accuracy of semantic-based metrics such as Sentence Transformers and BERT embedding cosine similarity. Since these metrics do not require third-party external APIs, have lower hardware requirements, and are not time-consuming, many may prefer to use them for automatic evaluations. The best accuracies were achieved by GPT-4-o and Llama 3.1 70B, both of which were used as judges. Llama 3.1 70B and 8B could be excellent choices for automatic evaluation, as they demonstrated no significant performance differences compared to open-source models and commercial ones in Open-domain QA.

Our experiments show the high accuracy of the lexical matching and semantic-based metrics applied to the short-form versions of the Natural Questions and TriviaQA datasets. The best performance in evaluating correct answers was achieved using Exact Match and BLEU Score, both with 100% accuracy. These metrics are simple, cost-effective, and easy to implement. Interestingly, when we applied our methodology (Layer 1 filtering), described in Section 3.4.2, to extract model-generated short answers from the Natural Questions and TriviaQA datasets, we observed notable results. Specifically, 40.4% of the Natural Questions data was evaluated with 99% accuracy, and 65.7% of the TriviaQA data was filtered with 100% accuracy. This was achieved simply by converting long answers into short ones and evaluating them using basic lexical matching metrics. Both Natural Questions and TriviaQA are widely used benchmarks in open-domain QA. The detailed results can be found in Appendix D, while Appendix E presents the impact of context length on both lexical matching and semantic-based metrics. Table 2 presents important results. These results are based on the short answer form. In the setups described in

Section 4.2, we used Exact Match and BLEU Score for Layer 1 filtering and only employed three metrics in Layer 2 voting. Model-generated answers from Natural Questions and TriviaQA were converted to short form. Layer 1 filtering was applied in all setups except the Only LLMs setup, which did not include Layer 1 filtering. The results indicate that using lexical matching metrics for Layer 2 can achieve evaluation accuracy comparable to GPT-3.5 Turbo, as shown in Table 1. Incorporating semantic-based metrics slightly improved the results, while adding only an open-source Llama LLM in Layer 2 yielded better results. Using only the Llama LLM 8B, which has lower hardware requirements, produced better results than the second setup but was weaker than the third setup. For setups using only LLMs without Layer 1 filtering, results for the Natural Questions dataset were even weaker than in the second setup.

Using Eq. 7 to calculate metrics for Layer 2, a single LLM in Layer 2 showed minimal differences compared to the third setup. In the seventh setup, all available metrics for Layer 2, selected using Eq. 7, required some human-annotated data to customize the metrics for the dataset. The latest setup, which did not involve dataset-specific customizations and used overall performance metrics selected for those without human-annotated data chosen by Equation Eq. 7, along with Recall, Llama 3.1 70B, and GPT-4-o in the voting layer. This achieved 3% better results for Natural Questions in short form, with 87% accuracy and 1% better for TriviaQA in short form, with 97% accuracy compared to GPT-4-o, which had been the best evaluation metric for both datasets. Interestingly, using only lexical matching and semantic-based metrics that do not require strong hardware or high costs resulted in accuracy just 4% lower than the best possible setup for automatic evaluation using well-known metrics and LLMs.

5 Conclusions

In this paper, we demonstrated that our fused approach, which utilizes our proposed formula for metrics selection and combines lexical-based metrics, semantic-based metrics, and LLMs as judges, achieves strong performance in the automatic evaluation of open-domain QA datasets. We also highlighted the often-overlooked effectiveness of lexical matching metrics, which perform well in evaluating short answers. This is particularly true given that many generated model answers are lengthy; our approach, which extracts short answers from these long responses, significantly improved evaluation results using these simple, low-computation metrics. Furthermore, our best evaluation setup, guided by our proposed formula, outperformed GPT-4-o, previously considered the top performer. Future work will focus on developing automated evaluation methods for Open-domain QA tasks that involve datasets without reference answers.

6 Limitations

Our methodology relies on publicly available pre-trained models as metrics. While these models perform well on general datasets, they may not be optimal for domain-specific contexts. Additionally, many of these models are trained primarily on English-language data, limiting their effectiveness for low-resource languages.

Furthermore, our testing was limited to the Natural Questions and TriviaQA datasets, which are well-established benchmarks in open-domain QA tasks. Incorporating a broader range of datasets could provide more comprehensive results and enhance diversity. The choice of datasets was influenced by the availability of publicly annotated human evaluations. Access to more human-annotated datasets in this domain would likely improve the diversity and robustness of the evaluation results and also the effectiveness of short answer extraction in lexical-based metrics is related to whether a gold answer appears within a longer answer. If the gold answer is not present but the long answer is correct, the short answer extraction methodology may not be useful. It appears that our dataset primarily includes gold answers within model-generated long answers for correct responses.

References

Vaibhav Adlakha, Parishad BehnamGhader, Xing Han Lu, Nicholas Meade, and Siva Reddy. 2024. Evaluating correctness and faithfulness of instruction-following models for question answering. *Transactions of the Association for Computational Linguistics*, 12:681–699.

Chenxin An, Shansan Gong, Ming Zhong, Xingjian Zhao, Mukai Li, Jun Zhang, Lingpeng Kong, and Xipeng Qiu. 2024. L-eval: Instituting standardized evaluation for long context language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long*

Papers), pages 14388–14411, Bangkok, Thailand. Association for Computational Linguistics.

Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondrej Dusek. 2024. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source LLMs. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 67–93, St. Julian's, Malta. Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

Xiaoyang Chen, Ben He, Hongyu Lin, Xianpei Han, Tianshu Wang, Boxi Cao, Le Sun, and Yingfei Sun. 2024. Spiral of silence: How is large language model killing information retrieval?—a case study on open domain question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14930–14951, Bangkok, Thailand. Association for Computational Linguistics.

Zheng Chu, Jingchang Chen, Qianglong Chen, Haotian Wang, Kun Zhu, Xiyuan Du, Weijiang Yu, Ming Liu, and Bing Qin. 2024. Beamaggr: Beam aggregation reasoning over multi-source knowledge for multi-hop question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1229–1248, Bangkok, Thailand. Association for Computational Linguistics.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The power of noise: Redefining retrieval for rag systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 719–729.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, and ... Amy Yang. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Prayushi Faldu, Indrajit Bhattacharya, and Mausam . 2024. Retinaqa: A robust knowledge base question answering model for both answerable and unanswerable questions. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6643–6656, Bangkok, Thailand. Association for Computational Linguistics.

Hung-Chieh Fang, Kuo-Han Hung, Chen-Wei Huang, and Yun-Nung Chen. 2022. Open-domain conversational question answering with historical answers. In *Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022*, pages 319–326, Online only. Association for Computational Linguistics.

Yufei Huang, Xu Han, and Maosong Sun. 2024. Fastfid: Improve inference efficiency of open domain question answering via sentence selection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6262–6276, Bangkok, Thailand. Association for Computational Linguistics.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023.acl-long.307:5591–5606.

M. G. Kendall. 1945. The treatment of ties in ranking problems. *Biometrika*, 33(3):239–251.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2024a. Pre-trained language models for text generation: A survey. *ACM Computing Surveys*, 56(9):1–39.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2024b. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36.

Xibo Li, Bowei Zou, Yifan Fan, Yanling Li, Ai Ti Aw, and Yu Hong. 2023. Interview evaluation: A novel approach for automatic evaluation of conversational question answering models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3435–3446, Singapore. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Adian Liusie, Potsawee Manakul, and Mark Gales. 2024. LLM comparative assessment: Zero-shot NLG evaluation through pairwise comparisons using large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 139–151, St. Julian's, Malta. Association for Computational Linguistics.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, and ... Shyamal Anadkat. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. Kilt: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Julian Risch, Timo Möller, Julian Gutsch, and Malte Pietsch. 2021. Semantic answer similarity for evaluating question answering models. In *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*, pages 149–157, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tobias Schimanski, Jingwei Ni, Mathias Kraus, Elliott Ash, and Markus Leippold. 2024. Towards faithful and robust llm specialists for evidence-based question-answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1913–1931, Bangkok, Thailand. Association for Computational Linguistics.

Hwanjun Song, Hang Su, Igor Shalyminov, Jason Cai, and Saab Mansour. 2024. FineSurE: Fine-grained summarization evaluation using LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 906–922, Bangkok, Thailand. Association for Computational Linguistics.

Cunxiang Wang, Sirui Cheng, Qipeng Guo, Yuanhao Yue, Bowen Ding, Zhikun Xu, Yidong Wang, Xiangkun Hu, Zheng Zhang, and Yue Zhang. 2023a. Evaluating open-qa evaluation. In *Advances in Neural Information Processing Systems*, volume 36, pages 77013–77042. Curran Associates, Inc.

Wenya Wang, Vivek Srikumar, Hannaneh Hajishirzi, and Noah A. Smith. 2023b. Elaboration-generating commonsense question answering at scale. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1619–1635, Toronto, Canada. Association for Computational Linguistics.

Ziang Xiao, Susu Zhang, Vivian Lai, and Q. Vera Liao. 2023. Evaluating evaluation metrics: A framework for analyzing NLG evaluation metrics using measurement theory. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10967–10982, Singapore. Association for Computational Linguistics.

Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Hu. 2024. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data*, 18(6):1–32.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with bertserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 72–77, Minneapolis, Minnesota. Association for Computational Linguistics.

Gal Yona, Roee Aharoni, and Mor Geva. 2024. Narrowing the knowledge evaluation gap: Open-domain question answering with multi-granularity answers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1:*

Long Papers), pages 6737–6751, Bangkok, Thailand. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

Liu Zhuang, Wayne Lin, Ya Shi, and Jun Zhao. 2021. A robustly optimized bert pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

A Correlation Matrix of Metrics

We explain the reason behind choosing $\Theta_{\text{min}} = 0.6$ and $\Theta_{\text{max}} = 0.9$. In the correlation matrix, the correlation between metrics is displayed. Metrics with a correlation below Θ_{min} (i.e., 0.6) generally exhibit unacceptable accuracy. Additionally, there are not many metrics with a correlation of 0.9 or higher. Even if such metrics are present, they are not ideal candidates for voting as their high correlation may reduce the benefit of including them. Instead, it is preferable to select other highly accurate metrics with lower correlation for voting purposes.

We calculate the correlations between LLMs for the Natural Questions and TriviaQA datasets separately. These correlations are shown when model-generated responses are converted into short form, as depicted in Figures 4 and 5, respectively. Additionally, we compute the correlations between lexical and semantic-based metrics, which are illustrated in Figures 6 and 7 for these datasets.

B Detailed Formula Calculation in Our Experiments

This is the general form of our formula, as discussed in Eq. 7, We will now provide more details on the calculations used in our experiments.

$$
\begin{aligned}
S = \arg \max_{S \subset \mathbf{M}, |S| \text{ odd}} \Bigg[& \beta \Bigg(\sum_{f_{\mathbf{M}_i} \in S} Acc(f_{\mathbf{M}_i}) \Bigg) \\
& - \lambda \Bigg(\sum_{\substack{f_{\mathbf{M}_i}, f_{\mathbf{M}_j} \in S \\ i<j}} \frac{1}{k} \sum_{l=1}^{k} \rho_l(f_{\mathbf{M}_i}, f_{\mathbf{M}_j}) \Bigg) \\
& + \gamma \Bigg(\sum_{f_{\mathbf{M}_i} \in S} \frac{1}{k} \sum_{l=1}^{k} \rho_l(f_{\mathbf{M}_i}, \mathbb{H}_i) \Bigg) \Bigg]
\end{aligned}
\tag{10}
$$

where

$$
\begin{gathered}
\Theta_{\text{min}} \leq \sum_{l=1}^{k} \rho_l(f_{\mathbf{M}_i}, f_{\mathbf{M}_j}) \leq \Theta_{\text{max}} \\
\text{for all } f_{\mathbf{M}_i}, f_{\mathbf{M}_j} \in S.
\end{gathered}
\tag{11}
$$

In our experiments, we utilized three correlation coefficients: Spearman's ρ_S, Kendall's τ, and Pearson's ρ_P. A voting system with 3 voters, one for each correlation method, was used to select the best metrics by maximizing the combined rankings of ρ_S, τ, and ρ_P for pairs $f_{\mathbf{M}_i}, f_{\mathbf{M}_j}$ and metrics with the human evaluation $\mathbb{H}_i$.

$$
\begin{aligned}
S = \arg \max_{S \subset \mathbf{M}, |S| \text{ odd}} \Bigg[& \sum_{f_{\mathbf{M}_i} \in S} Acc(f_{\mathbf{M}_i}) \\
& - \lambda \Bigg(\sum_{\substack{f_{\mathbf{M}_i}, f_{\mathbf{M}_j} \in S \\ i<j}} \frac{1}{3} \Big(\rho_S(f_{\mathbf{M}_i}, f_{\mathbf{M}_j}) \\
& \quad + \tau(f_{\mathbf{M}_i}, f_{\mathbf{M}_j}) + \rho_P(f_{\mathbf{M}_i}, f_{\mathbf{M}_j}) \Big) \Bigg) \\
& + \gamma \Bigg(\sum_{f_{\mathbf{M}_i} \in S} \frac{1}{3} \Big(\rho_S(f_{\mathbf{M}_i}, \mathbb{H}_i) \\
& \quad + \tau(f_{\mathbf{M}_i}, \mathbb{H}_i) + \rho_P(f_{\mathbf{M}_i}, \mathbb{H}_i) \Big) \Bigg) \Bigg]
\end{aligned}
\tag{12}
$$

In our case, we considered Θ_{min} and Θ_{max} to be 0.6 and 0.9, respectively, with

$$
\begin{aligned}
\Theta_{\text{min}} \leq & \frac{1}{3} \Big(\rho_S(f_{\mathbf{M}_i}, f_{\mathbf{M}_j}) \\
& + \tau(f_{\mathbf{M}_i}, f_{\mathbf{M}_j}) + \rho_P(f_{\mathbf{M}_i}, f_{\mathbf{M}_j}) \Big) \\
\leq & \Theta_{\text{max}} \\
& \text{for all } f_{\mathbf{M}_i}, f_{\mathbf{M}_j} \in S.
\end{aligned}
\tag{13}
$$

In the formulas, S is an odd-sized subset of metrics $\mathbf{M}$. The parameter β scales the sum of metric values $Acc(f_{\mathbf{M}_i})$ within S. With $k = 3$ representing the number of correlations used. For Spearman's correlation, $d_{ij,n}$ is the rank difference and N is the number of pairs. Kendall's tau uses concord and discord for concordant and discordant pairs, respectively, with N_{ij} as the number of pairs compared. Pearson's correlation involves $x_{ij,n}$ and $y_{ij,n}$ as data points, $\bar{x}_{ij}$ and $\bar{y}_{ij}$ as their means, and N_{iH} as the number of pairs between $f_{\mathbf{M}_i}$ and $\mathbb{H}_i$. We can expand the formula as follows:

$$
\begin{aligned}
S = \arg \max_{S \subset \mathbf{M}, |S| \text{ odd}} & \Bigg[\beta \Bigg(\sum_{f_{\mathbf{M}_i} \in S} Acc(f_{\mathbf{M}_i}) \Bigg) \\
& - \lambda \Bigg(\sum_{\substack{f_{\mathbf{M}_i}, f_{\mathbf{M}_j} \in S \\ i<j}} \frac{1}{3} \Bigg(1 - \frac{6 \sum_{n=1}^{N} d_{ij,n}^2}{N(N^2-1)} \\
& + \frac{\text{concord}(f_{\mathbf{M}_i}, f_{\mathbf{M}_j}) - \text{discord}(f_{\mathbf{M}_i}, f_{\mathbf{M}_j})}{\frac{1}{2} N_{ij}(N_{ij}-1)} \\
& + \frac{\sum_{n=1}^{N_{ij}} (x_{ij,n} - \bar{x}_{ij})(y_{ij,n} - \bar{y}_{ij})}{\sqrt{\sum_{n=1}^{N_{ij}} (x_{ij,n} - \bar{x}_{ij})^2 \sum_{n=1}^{N_{ij}} (y_{ij,n} - \bar{y}_{ij})^2}} \Bigg) \Bigg) \\
& + \gamma \Bigg(\sum_{f_{\mathbf{M}_i} \in S} \frac{1}{3} \Bigg(1 - \frac{6 \sum_{n=1}^{N} d_{iH,n}^2}{N(N^2-1)} \\
& + \frac{\text{concord}(f_{\mathbf{M}_i}, \mathbb{H}_i) - \text{discord}(f_{\mathbf{M}_i}, \mathbb{H}_i)}{\frac{1}{2} N_{iH}(N_{iH}-1)} \\
& + \frac{\sum_{n=1}^{N_{iH}} (x_{iH,n} - \bar{x}_{iH})(y_{iH,n} - \bar{y}_{iH})}{\sqrt{\sum_{n=1}^{N_{iH}} (x_{iH,n} - \bar{x}_{iH})^2 \sum_{n=1}^{N_{iH}} (y_{iH,n} - \bar{y}_{iH})^2}} \Bigg) \Bigg) \Bigg]
\end{aligned} \tag{14}
$$

C Prompts for LLM to Act as a Metric Scorer

The following prompts instruct the LLM to act as a metric scorer, evaluating the correctness of predicted answers on a scale from 0 to 1. Depicted in 2 and 3

> Given the question: '{question}', the predicted answer: '{answer}', and the correct gold answer: '{gold_answer}', score the predicted answer from 0 to 1 based on its correctness and similarity to the gold answer. Just return the score in the format: score:<value>

Figure 2: Prompt used to instruct the LLM to score the predicted answer from 0 to 1 based on its correctness and similarity to the provided gold answer.

D Filtering Metrics Threshold Accuracy Comparison

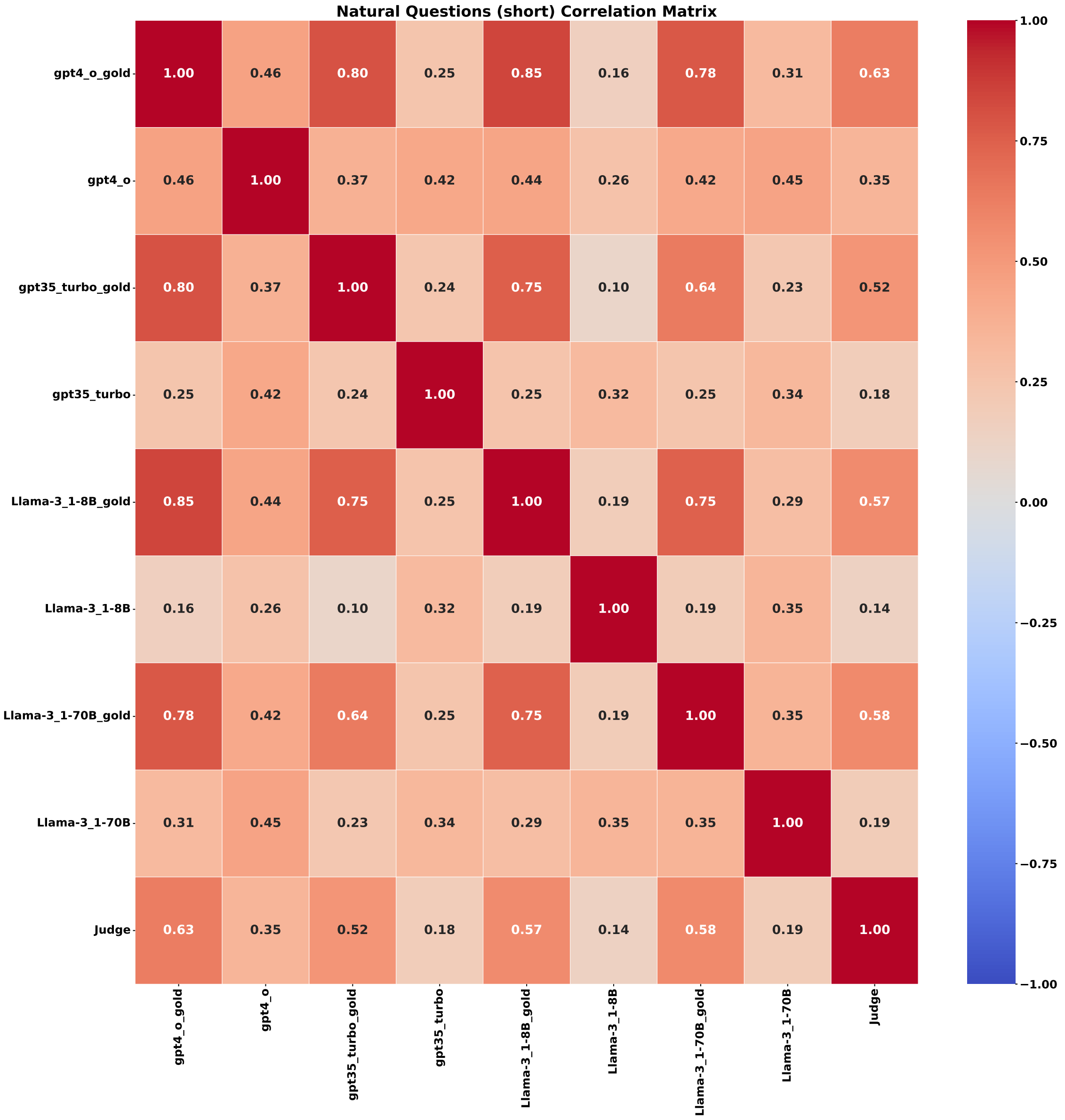
Table 3 presents the accuracy comparison between automated evaluation metrics and human judgments. A threshold T_M of 0.5 was used in our experiments.The accuracy is calculated by true positive $f_M >= T_M$ and true negative $f_M < T_M$

> Given the question: '{question}' and the predicted answer: '{answer}', score the predicted answer from 0 to 1 based on its correctness. Just return the score in the format: score:<value>

Figure 3: Prompt used to instruct the LLM to score the predicted answer from 0 to 1 based on its correctness, without access to a gold answer.

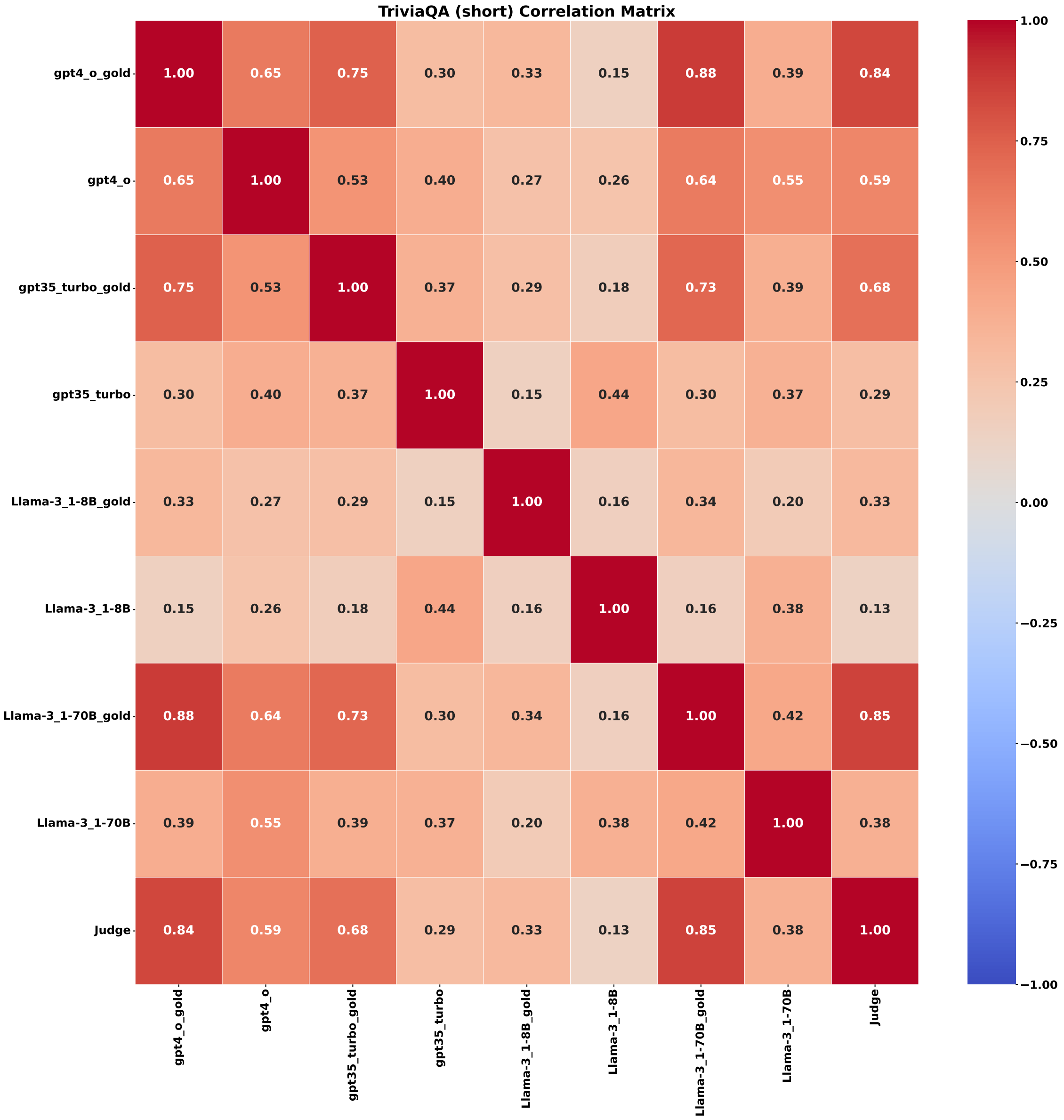
E Effect of Context Length on Lexical Matching and Semantic-Based Metrics Accuracy

We present an analysis of the effect of converting model-generated answers into short-form versus long-form on accuracy. Specifically, we compare the accuracy of lexical matching and semantic-based metrics when applied to both short and long answers. As shown in Figure 8



	gpt4_o_gold	gpt4_o	gpt35_turbo_gold	gpt35_turbo	Llama-3_1-8B_gold	Llama-3_1-8B	Llama-3_1-70B_gold	Llama-3_1-70B	Judge
gpt4_o_gold	1.00	0.46	0.80	0.25	0.85	0.16	0.78	0.31	0.63
gpt4_o	0.46	1.00	0.37	0.42	0.44	0.26	0.42	0.45	0.35
gpt35_turbo_gold	0.80	0.37	1.00	0.24	0.75	0.10	0.64	0.23	0.52
gpt35_turbo	0.25	0.42	0.24	1.00	0.25	0.32	0.25	0.34	0.18
Llama-3_1-8B_gold	0.85	0.44	0.75	0.25	1.00	0.19	0.75	0.29	0.57
Llama-3_1-8B	0.16	0.26	0.10	0.32	0.19	1.00	0.19	0.35	0.14
Llama-3_1-70B_gold	0.78	0.42	0.64	0.25	0.75	0.19	1.00	0.35	0.58
Llama-3_1-70B	0.31	0.45	0.23	0.34	0.29	0.35	0.35	1.00	0.19
Judge	0.63	0.35	0.52	0.18	0.57	0.14	0.58	0.19	1.00

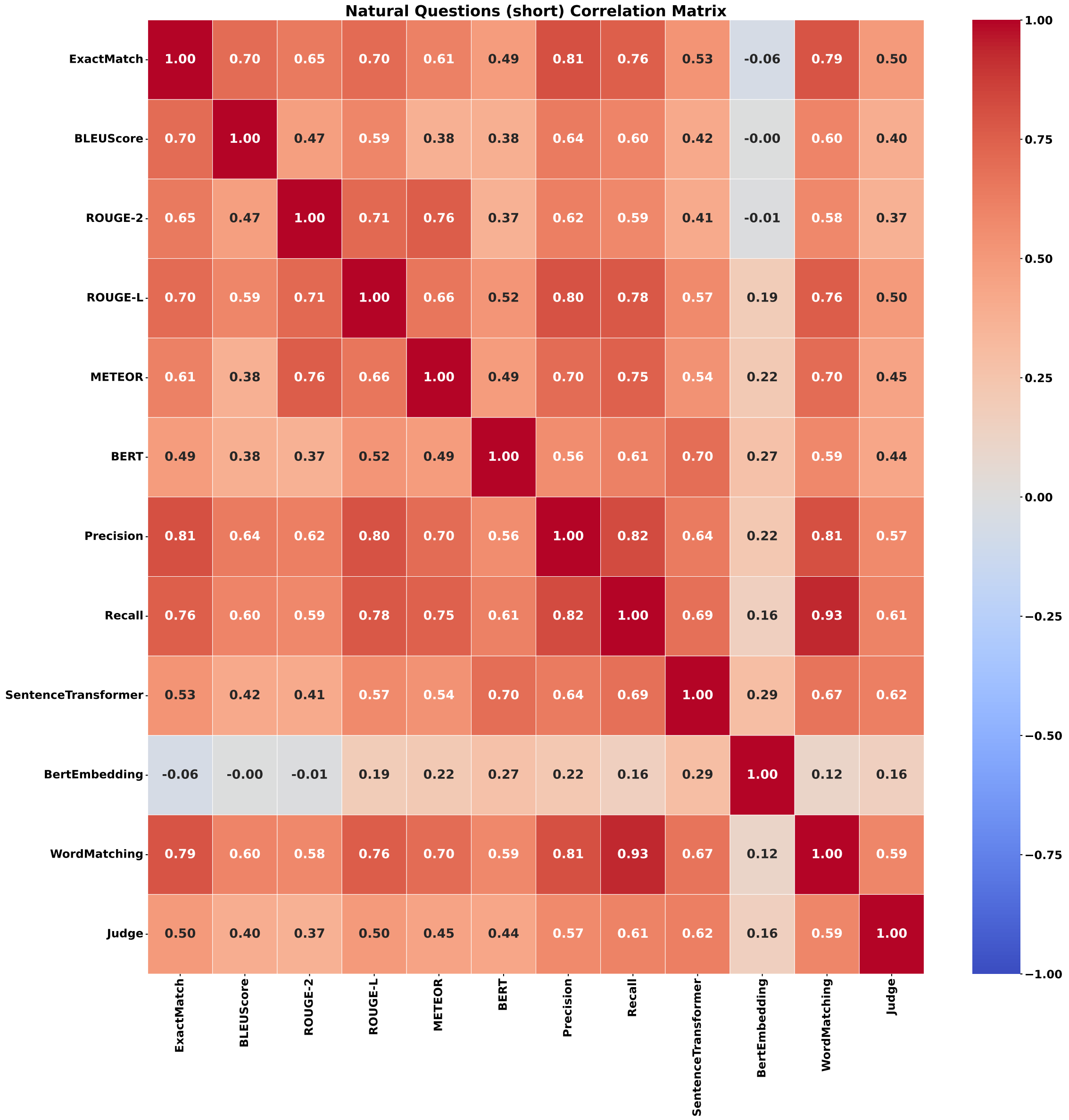
Figure 4: Correlation matrix for the Natural Questions, LLMs metrics. Judge is human evaluation.



TriviaQA (short) Correlation Matrix

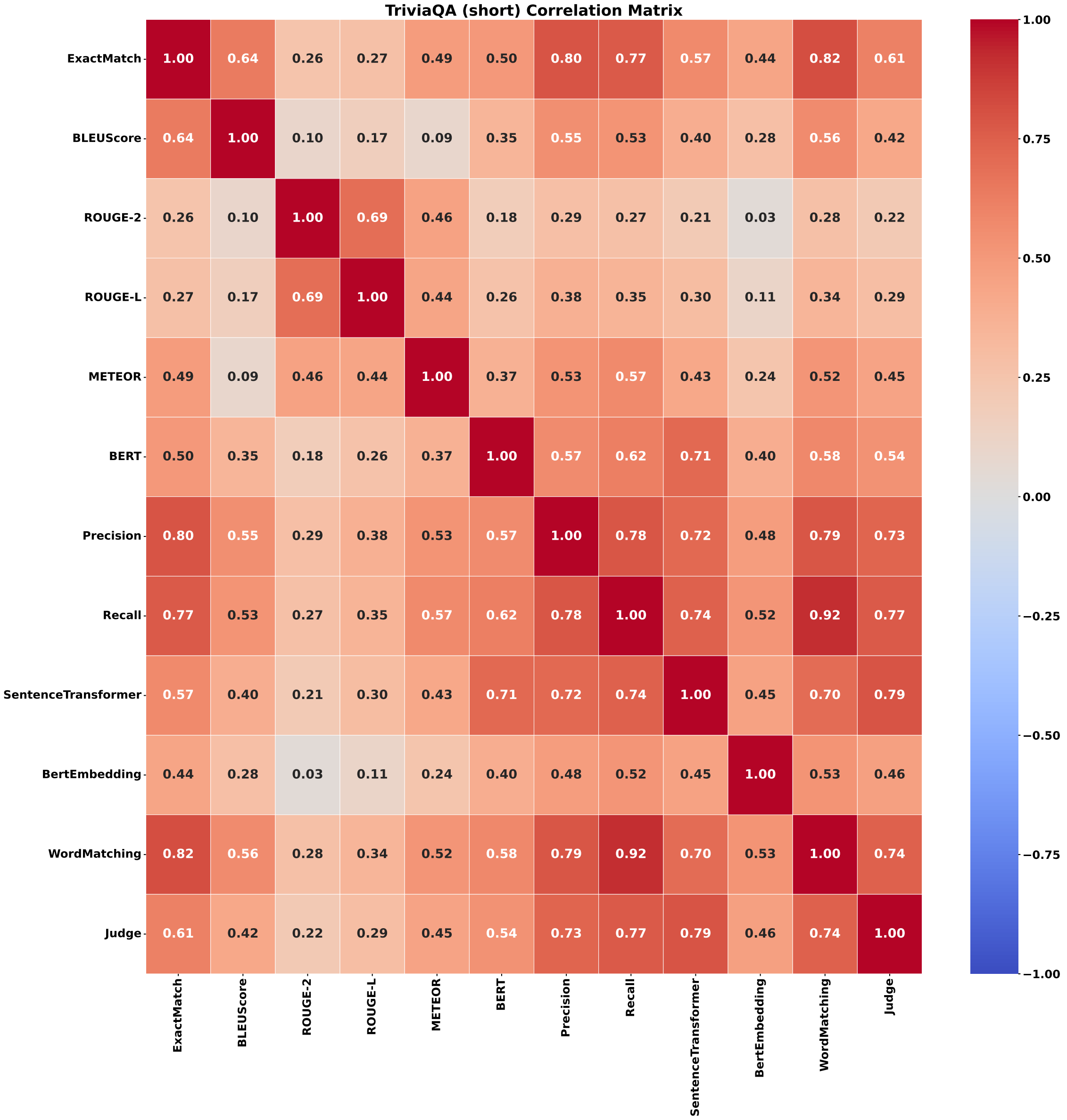
	gpt4_o_gold	gpt4_o	gpt35_turbo_gold	gpt35_turbo	Llama-3_1-8B_gold	Llama-3_1-8B	Llama-3_1-70B_gold	Llama-3_1-70B	Judge
gpt4_o_gold	1.00	0.65	0.75	0.30	0.33	0.15	0.88	0.39	0.84
gpt4_o	0.65	1.00	0.53	0.40	0.27	0.26	0.64	0.55	0.59
gpt35_turbo_gold	0.75	0.53	1.00	0.37	0.29	0.18	0.73	0.39	0.68
gpt35_turbo	0.30	0.40	0.37	1.00	0.15	0.44	0.30	0.37	0.29
Llama-3_1-8B_gold	0.33	0.27	0.29	0.15	1.00	0.16	0.34	0.20	0.33
Llama-3_1-8B	0.15	0.26	0.18	0.44	0.16	1.00	0.16	0.38	0.13
Llama-3_1-70B_gold	0.88	0.64	0.73	0.30	0.34	0.16	1.00	0.42	0.85
Llama-3_1-70B	0.39	0.55	0.39	0.37	0.20	0.38	0.42	1.00	0.38
Judge	0.84	0.59	0.68	0.29	0.33	0.13	0.85	0.38	1.00

Figure 5: Correlation matrix for the TriviaQA, LLMs metrics. Judge is human evaluation.



	ExactMatch	BLEUScore	ROUGE-2	ROUGE-L	METEOR	BERT	Precision	Recall	SentenceTransformer	BertEmbedding	WordMatching	Judge
ExactMatch	1.00	0.70	0.65	0.70	0.61	0.49	0.81	0.76	0.53	-0.06	0.79	0.50
BLEUScore	0.70	1.00	0.47	0.59	0.38	0.38	0.64	0.60	0.42	-0.00	0.60	0.40
ROUGE-2	0.65	0.47	1.00	0.71	0.76	0.37	0.62	0.59	0.41	-0.01	0.58	0.37
ROUGE-L	0.70	0.59	0.71	1.00	0.66	0.52	0.80	0.78	0.57	0.19	0.76	0.50
METEOR	0.61	0.38	0.76	0.66	1.00	0.49	0.70	0.75	0.54	0.22	0.70	0.45
BERT	0.49	0.38	0.37	0.52	0.49	1.00	0.56	0.61	0.70	0.27	0.59	0.44
Precision	0.81	0.64	0.62	0.80	0.70	0.56	1.00	0.82	0.64	0.22	0.81	0.57
Recall	0.76	0.60	0.59	0.78	0.75	0.61	0.82	1.00	0.69	0.16	0.93	0.61
SentenceTransformer	0.53	0.42	0.41	0.57	0.54	0.70	0.64	0.69	1.00	0.29	0.67	0.62
BertEmbedding	-0.06	-0.00	-0.01	0.19	0.22	0.27	0.22	0.16	0.29	1.00	0.12	0.16
WordMatching	0.79	0.60	0.58	0.76	0.70	0.59	0.81	0.93	0.67	0.12	1.00	0.59
Judge	0.50	0.40	0.37	0.50	0.45	0.44	0.57	0.61	0.62	0.16	0.59	1.00

Figure 6: Correlation matrix for the Natural Questions, lexical and semantic-based metrics. Judge is human evaluation.



TriviaQA (short) Correlation Matrix

	ExactMatch	BLEUScore	ROUGE-2	ROUGE-L	METEOR	BERT	Precision	Recall	SentenceTransformer	BertEmbedding	WordMatching	Judge
ExactMatch	1.00	0.64	0.26	0.27	0.49	0.50	0.80	0.77	0.57	0.44	0.82	0.61
BLEUScore	0.64	1.00	0.10	0.17	0.09	0.35	0.55	0.53	0.40	0.28	0.56	0.42
ROUGE-2	0.26	0.10	1.00	0.69	0.46	0.18	0.29	0.27	0.21	0.03	0.28	0.22
ROUGE-L	0.27	0.17	0.69	1.00	0.44	0.26	0.38	0.35	0.30	0.11	0.34	0.29
METEOR	0.49	0.09	0.46	0.44	1.00	0.37	0.53	0.57	0.43	0.24	0.52	0.45
BERT	0.50	0.35	0.18	0.26	0.37	1.00	0.57	0.62	0.71	0.40	0.58	0.54
Precision	0.80	0.55	0.29	0.38	0.53	0.57	1.00	0.78	0.72	0.48	0.79	0.73
Recall	0.77	0.53	0.27	0.35	0.57	0.62	0.78	1.00	0.74	0.52	0.92	0.77
SentenceTransformer	0.57	0.40	0.21	0.30	0.43	0.71	0.72	0.74	1.00	0.45	0.70	0.79
BertEmbedding	0.44	0.28	0.03	0.11	0.24	0.40	0.48	0.52	0.45	1.00	0.53	0.46
WordMatching	0.82	0.56	0.28	0.34	0.52	0.58	0.79	0.92	0.70	0.53	1.00	0.74
Judge	0.61	0.42	0.22	0.29	0.45	0.54	0.73	0.77	0.79	0.46	0.74	1.00

Figure 7: Correlation matrix for the TriviaQA, lexical and semantic-based metrics. Judge is human evaluation.

Metric	**Natural Questions**		**TriviaQA**	
	$f_M >= T_M$	$f_M < T_M$	$f_M >= T_M$	$f_M < T_M$
Exact Match	**1.00**	0.47	**1.00**	0.49
BLEU Score	0.99	0.41	**1.00**	0.33
METEOR Score	0.96	0.60	0.99	0.72
ROUGE-2	0.99	0.40	**1.00**	0.23
ROUGE-L	0.96	0.52	0.98	0.28
Word Matching	0.95	0.62	0.98	0.68
Precision	0.95	0.60	0.98	0.73
Recall	0.95	0.64	0.98	0.75
BERT Score	0.83	0.62	0.90	0.72
Sentence Transformer	0.90	0.71	0.95	0.88
BERT Embedding	**0.95**	0.62	**0.99**	0.72
GPT-4-o (Gold)	0.89	0.69	**0.97**	0.89
GPT-4-o	0.79	0.58	0.91	0.78
GPT-3.5 Turbo (Gold)	0.87	0.65	0.93	0.75
GPT-3.5 Turbo	0.74	0.59	0.86	0.46
Meta-Llama 3.1 70B (Gold)	0.89	0.69	**0.97**	0.90
Meta-Llama 3.1 70B	0.75	0.50	0.88	0.52
Meta-Llama 3.1 8B (Gold)	**0.94**	0.60	0.93	0.30
Meta-Llama 3.1 8B	0.73	0.41	0.84	0.31

Table 3: This table presents the accuracy of various evaluation metrics applied above and below their thershold from two datasets: Natural Questions and TriviaQA. The metrics include both lexical-based and semantic-based methods, as well as using LLMs as judges, similar to a metric. Results are further divided based on whether gold answers were provided to the LLMs (denoted by "(Gold)") or not. The best-performing results in each group are bolded. The overall best accuracy for each dataset is bolded and underlined

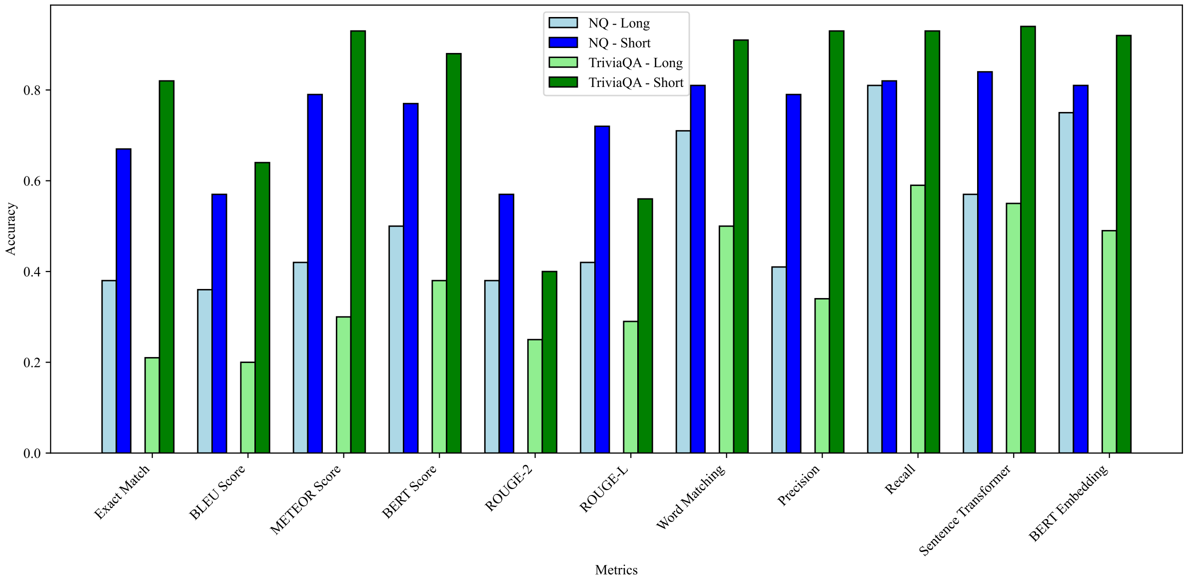


Figure 8: The figure illustrates the performance accuracy of various metrics in comparison to human judgments. It compares the accuracy of model-generated answers in both their short and long forms. Results for the Natural Questions (NQ) dataset are represented in blue, while the TriviaQA dataset is represented in green.